48760

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Application of : **ARIDOR et al.**

Serial No.: 10/634,319	: Group Art Unit: 2163
Filed : August 1, 2003	: Examiner: Hanh B. Thai

For : INFORMATION SEARCH USING KNOWLEDGE AGENTS

Honorable Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

DECLARATION UNDER 37 CFR 1.131

Sir:

We, the undersigned, Yariv Aridor, David Carmel, Michael Herscovici, Yoelle Maarek-Smadja, Aya Soffer and Ronny Lempel, hereby declare as follows:

1) We are the Applicants in the patent application identified above, and are the inventors of the subject matter described and claimed in claims 35-62 therein.

2) Prior to February 25, 2000, we reduced our invention to practice, as described and claimed in the subject application, in Israel, a WTO country. We implemented the invention in the form of software code in the Java programming language, and then tested the code successfully in a prototype system.

3) As evidence of the reduction to practice of the present invention, we attach hereto in Exhibits A-C parts of the Java source code that we used to implement the invention:

- Exhibit A: Class KnowledgeAgents.Agent
- Exhibit B: Class KnowledgeAgents.AgentGUI
- Exhibit C: Class KnowledgeAgents.Repository

A directory listing in Exhibit D (generated by the file archiving system used in the IBM Haifa Research Laboratory) shows the date on which the above source code files were stored on disk. The dates of the files, which are blacked out in Exhibit D, are prior to February 25, 2000. Results of testing this code are reported in a paper we prepared prior to February 25, 2000, entitled, "Knowledge Agents on the Web," which is attached hereto as Exhibit E.

4) Generally speaking, the software code in Exhibits A-C performs the functions of searching a corpus of documents, such as the World Wide Web, using knowledge agents that have developed specializations in certain knowledge domains. The following table shows the correspondence between the elements of method claims 35-48 in the present patent application and elements of the source code in Exhibits A-C:

Claim 35	**Source code**
A method for searching a corpus of documents, comprising:	KnowledgeAgents.AgentGUI and KnowledgeAgents.Agent provide a set of APIs to construct a domain specific agent and to use it for domain-specific search. See, for example, Exhibit A, lines 290-305.

defining a knowledge domain;	The KnowledgeAgents.Agent constructor (Exhibit A, lines 184-222) constructs a new agent for a given domain. The domain is defined by certain Web sites held in a repository (Exhibit C) that is specific to the agent/domain.
identifying a set of reference documents in the corpus pertinent to the domain;	The user initially specifies a set of sites (reference documents) using the “Add Sites” command in KnowledgeAgents.AgentGUI.actionPerformed() (Exhibit B, lines 445 - 450). This code calls the ActionAddSites sub-class defined in KnowledgeAgents.Agent (Exhibit A, lines 144 - 165), which adds the sites to the agent repository.
searching the corpus using the set of reference documents to find one or more of the documents in the corpus that contain information in the domain relevant to a first query; and	KnowledgeAgents.Agent.textQuery() (Exhibit A, lines 289-399) searches for a specific query over the agent’s domain and finds sites that satisfy the query. The Boolean parameter *update* of this method (line 316) controls whether the search results will be added/updated into the agent’s repository.

adding at least one of the found documents to the set of reference documents for use in searching the corpus for information in the domain relevant to a second, subsequent query, which is substantially different from the first query.	KnowledgeAgents.Agent.rankSites() (Exhibit A, lines 539-686) ranks the sites in which search results were found in the query stage. The repository is updated (lines 685-686) to include the best sites, i.e., sites with the highest weights. The updated repository is then available for use the next time textQuery() is invoked.
Claim 36	
The method according to claim 35, wherein inputting the first query comprises inputting one or more search terms.	The user inputs query terms as text strings using the “Refine Query” command in KnowledgeAgents.AgentGUI. actionPerformed() (Exhibit B, lines 436-441).

Claim 37	
The method according to claim 36, wherein searching the corpus comprises finding lexical characteristics of terms in the reference documents and refining the search terms using the lexical characteristics.	KnowledgeAgents.Agent.refineQuery() (Exhibit A, lines 1068-1087) receives as input a given query and number of terms to use in expanding the query. This method then expands the query (i.e., refines the search terms) using lexical characteristics (in the form of lexical affinities - "LAs") extracted from the agent's repository.
Claim 38	
The method according to claim 35, wherein inputting the first query comprises specifying one or more documents representative of the information to be found in the corpus.	KnowledgeAgents.Agent.linkQuery() (Exhibit A, lines 400-451) receives as input a set of documents (sites) that the user has specified as representing the information available on the Web. (The method returns an expanded set of sites that are optimally relevant to the given sites.)

Claim 39	
The method according to claim 35, wherein searching the corpus comprises searching the corpus to find the documents that contain the information relevant to the query and ranking the found documents by comparing them to the set of reference documents.	KnowledgeAgents.Agent.textQuery() calls KnowledgeAgents.Agent.rankSites() (Exhibit A, lines 539-687) which ranks the sites returned by the search according to the sites in the agent's repository. The comparison may be based on textual resemblance or on links, as noted with respect to claims 40 and 41 below.
Claim 40	
The method according to claim 39, wherein ranking the found documents comprises evaluating a textual resemblance between the found documents and the reference documents.	KnowledgeAgents.Agent.rankSites() ranks the search results by evaluating the textual resemblance between documents. It performs this function by calling (at line 595) KnowledgeAgents.Agent.normalizeTextWeights() (Exhibit A, lines 1213-1235).

Claim 41	
The method according to claim 39, wherein ranking the found documents comprises assessing links between the found documents and the reference documents.	KnowledgeAgents.Agent.getForwSet() finds all pages that the pages in the result set link to (Exhibit A, lines 729-868), while KnowledgeAgents.Agent.getBackSet() finds all pages that link to the result set (lines 689-728). Both methods are called by KnowledgeAgents.Agent.textQuery() in order to find all pages linking to and linked by the result page.
Claim 42	
The method according to claim 39, wherein adding the at least one of the found documents comprises adding at least the document having the highest ranking.	In KnowledgeAgents.Agent.rankSites(), after collecting all pages and scoring them, the top scored documents are used to update the agent's repository (Exhibit A, lines 684 - 686) by calling KnowledgeAgents.Repository.transfuse() (Exhibit C, lines 197-297).

Claim 43	
The method according to claim 35, wherein adding the at least one of the found documents comprises removing one of the documents from the set responsive to adding the at least one of the found documents.	KnowledgeAgents.Repository.transfuse() "transfuses new good sites into the repository, replacing stale sites" (Exhibit C, line 198).
Claim 44	
The method according to claim 43, and comprising tracking a level of relevance of the reference documents to the queries, and wherein removing the one of the documents comprises removing one of the reference documents whose tracked level of relevance is low.	The class KnowledgeAgents.Repository tracks the relevance level for each of the sites in the repository in the SiteDB local object. (This object is declared in Exhibit C, line 17, and the scores are updated at lines 235-252.) These relevance levels are used to determine whether old sites will be kept or replaced during the update process performed by the KnowledgeAgents.Repository.transfuse() method, as described above.

Claim 45	
The method according to claim 35, wherein the corpus comprises at least a part of the World Wide Web, and the documents comprise Web pages, and wherein searching the corpus comprises conveying the query to one or more Web search engines.	KnowledgeAgents.Agent.textQuery() searches the World Wide Web (WWW, Exhibit A, lines 303-305). This method calls one or more search engines (line 315) to perform the search. In Exhibit E (page 16, last paragraph) we described the use of the AltaVista Web search engine in this manner.
Claim 46	
The method according to claim 45, wherein inputting the first query comprises receiving the query from a user of a pervasive device, and wherein searching the corpus comprises searching while the device is disconnected from the Web.	This feature may be implemented using the code in Exhibits A-C, but it is not explicitly shown in the exhibits. In regard to this claim, the Examiner indicated that Bowman et al. (U.S. Patent 6,006,225) would have led a person of ordinary skill in the art to carry out the step of searching while the device that received the search query is disconnected from the Web.

Claim 47	
The method according to claim 35, wherein identifying the set of reference documents comprises opening one or more files of a knowledge base on a computer in which data regarding the reference documents are saved.	This feature may be implemented using the code in Exhibits A-C, by applying the methods in the code to reference documents in a knowledge base on a computer rather than on the World Wide Web, but it is not explicitly shown in the exhibits. In regard to this claim, the Examiner indicated that Liddy et al. (U.S. Patent 6,304,864) would have led a person of ordinary skill in the art to open one or more files of a knowledge base on a computer in the context of identifying the set of reference documents as recited in this claim.

Claim 48	
The method according to claim 47, wherein identifying the set of reference documents comprises identifying the set of documents used by a first user in searching the corpus, and wherein opening the one or more files comprises copying the files for use by a second user in searching the corpus for information in the domain.	This feature may be implemented using the code in Exhibits A-C, by copying files opened by a first user for use by a second user, but it is not explicitly shown in the exhibits. In regard to this claim, the Examiner indicated that Liddy et al. (U.S. Patent 6,304,864) would have led a person of ordinary skill in the art to copy files opened by a first user for use by a second user in searching a corpus for information as recited in this claim.

5) Claims 51-57 and 59-61 recite apparatus and a computer software product, with limitations similar to those of certain of method claims 35-48. Based on the similarity of subject matter between the method, apparatus and software claims, it can similarly be demonstrated that we reduced to practice the entire invention recited in claims 51-57 and 59-61 prior to February 25, 2000.

6) We described the capabilities of our search software (as presented in Exhibits A-C) in the paper that is attached hereto as Exhibit E. As explained in section 4 of this paper, we defined knowledge agents in a number of different knowledge domains, including palm pilots, cryptography, artificial intelligence, geographic information systems, information

retrieval and Star Wars. The test results are described in detail on pages 14-18 of Exhibit E. The reported results demonstrate that our software successfully carried out the functions that are recited in the claims above.

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and conjecture are thought to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

Yariv Aridor
Citizen of Israel
31/b Yaalom Street, Zichron
Yaakov 30900
Israel

Date:
Jan 23, 2007

David Carmel
Citizen of Israel
12/5 Alexander Yanai Street,
Haifa 34816
Israel

Date:

US 10/634,319

Declaration under 37 C.F.R 1.131 by Aridor et al.

____

Michael Herscovici
Citizen of Israel
14 Got Levin Street, Haifa
32922
Israel

Date:

____

____

Yoelle Maarek-Smadja
Citizen of Israel
[Address] 57 Albert Schweitzer, Haifa 34995
Israel

Date:

____

____

Aya Soffer
Citizen of Israel
33 Disraeli Street, Haifa
34333
Israel

Date:

____

____

Ronny Lempel
Citizen of Israel
1 Moshe Sneh Street, Haifa
34987
Israel

Date:

____

48760

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Application of : **ARIDOR et al.**

Serial No.: 10/634,319 : Group Art Unit: 2163

Filed : August 1, 2003 : Examiner: Hanh B. Thai

For : INFORMATION SEARCH USING KNOWLEDGE AGENTS

Honorable Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

DECLARATION UNDER 37 CFR 1.131

Sir:

We, the undersigned, Yariv Aridor, David Carmel, Michael Herscovici, Yoelle Maarek-Smadja, Aya Soffer and Ronny Lempel, hereby declare as follows:

1) We are the Applicants in the patent application identified above, and are the inventors of the subject matter described and claimed in claims 35-62 therein.

2) Prior to February 25, 2000, we reduced our invention to practice, as described and claimed in the subject application, in Israel, a WTO country. We implemented the invention in the form of software code in the Java programming language, and then tested the code successfully in a prototype system.

3) As evidence of the reduction to practice of the present invention, we attach hereto in Exhibits A-C parts of the Java source code that we used to implement the invention:

- Exhibit A: Class KnowledgeAgents.Agent
- Exhibit B: Class KnowledgeAgents.AgentGUI
- Exhibit C: Class KnowledgeAgents.Repository

A directory listing in Exhibit D (generated by the file archiving system used in the IBM Haifa Research Laboratory) shows the date on which the above source code files were stored on disk. The dates of the files, which are blacked out in Exhibit D, are prior to February 25, 2000. Results of testing this code are reported in a paper we prepared prior to February 25, 2000, entitled, "Knowledge Agents on the Web," which is attached hereto as Exhibit E.

4) Generally speaking, the software code in Exhibits A-C performs the functions of searching a corpus of documents, such as the World Wide Web, using knowledge agents that have developed specializations in certain knowledge domains. The following table shows the correspondence between the elements of method claims 35-48 in the present patent application and elements of the source code in Exhibits A-C:

Claim 35	**Source code**
A method for searching a corpus of documents, comprising:	KnowledgeAgents.AgentGUI and KnowledgeAgents.Agent provide a set of APIs to construct a domain specific agent and to use it for domain-specific search. See, for example, Exhibit A, lines 290-305.

defining a knowledge domain;	The KnowledgeAgents.Agent constructor (Exhibit A, lines 184-222) constructs a new agent for a given domain. The domain is defined by certain Web sites held in a repository (Exhibit C) that is specific to the agent/domain.
identifying a set of reference documents in the corpus pertinent to the domain;	The user initially specifies a set of sites (reference documents) using the "Add Sites" command in KnowledgeAgents.AgentGUI.actionPerformed() (Exhibit B, lines 445 - 450). This code calls the ActionAddSites sub-class defined in KnowledgeAgents.Agent (Exhibit A, lines 144 - 165), which adds the sites to the agent repository.
searching the corpus using the set of reference documents to find one or more of the documents in the corpus that contain information in the domain relevant to a first query; and	KnowledgeAgents.Agent.textQuery() (Exhibit A, lines 289-399) searches for a specific query over the agent's domain and finds sites that satisfy the query. The Boolean parameter *update* of this method (line 316) controls whether the search results will be added/updated into the agent's repository.

adding at least one of the found documents to the set of reference documents for use in searching the corpus for information in the domain relevant to a second, subsequent query, which is substantially different from the first query.	KnowledgeAgents.Agent.rankSites() (Exhibit A, lines 539-686) ranks the sites in which search results were found in the query stage. The repository is updated (lines 685-686) to include the best sites, i.e., sites with the highest weights. The updated repository is then available for use the next time textQuery() is invoked.
Claim 36	
The method according to claim 35, wherein inputting the first query comprises inputting one or more search terms.	The user inputs query terms as text strings using the “Refine Query” command in KnowledgeAgents.AgentGUI. actionPerformed() (Exhibit B, lines 436-441).

Claim 37	
The method according to claim 36, wherein searching the corpus comprises finding lexical characteristics of terms in the reference documents and refining the search terms using the lexical characteristics.	KnowledgeAgents.Agent.refineQuery() (Exhibit A, lines 1068-1087) receives as input a given query and number of terms to use in expanding the query. This method then expands the query (i.e., refines the search terms) using lexical characteristics (in the form of lexical affinities - "LAs") extracted from the agent's repository.
Claim 38	
The method according to claim 35, wherein inputting the first query comprises specifying one or more documents representative of the information to be found in the corpus.	KnowledgeAgents.Agent.linkQuery() (Exhibit A, lines 400-451) receives as input a set of documents (sites) that the user has specified as representing the information available on the Web. (The method returns an expanded set of sites that are optimally relevant to the given sites.)

Claim 39	
The method according to claim 35, wherein searching the corpus comprises searching the corpus to find the documents that contain the information relevant to the query and ranking the found documents by comparing them to the set of reference documents.	KnowledgeAgents.Agent.textQuery() calls KnowledgeAgents.Agent.rankSites() (Exhibit A, lines 539-687) which ranks the sites returned by the search according to the sites in the agent's repository. The comparison may be based on textual resemblance or on links, as noted with respect to claims 40 and 41 below.
Claim 40	
The method according to claim 39, wherein ranking the found documents comprises evaluating a textual resemblance between the found documents and the reference documents.	KnowledgeAgents.Agent.rankSites() ranks the search results by evaluating the textual resemblance between documents. It performs this function by calling (at line 595) KnowledgeAgents.Agent. normalizeTextWeights() (Exhibit A, lines 1213-1235).

Claim 41	
The method according to claim 39, wherein ranking the found documents comprises assessing links between the found documents and the reference documents.	KnowledgeAgents.Agent.getForwSet() finds all pages that the pages in the result set link to (Exhibit A, lines 729-868), while KnowledgeAgents.Agent.getBackSet() finds all pages that link to the result set (lines 689-728). Both methods are called by KnowledgeAgents.Agent.textQuery() in order to find all pages linking to and linked by the result page.
Claim 42	
The method according to claim 39, wherein adding the at least one of the found documents comprises adding at least the document having the highest ranking.	In KnowledgeAgents.Agent.rankSites(), after collecting all pages and scoring them, the top scored documents are used to update the agent's repository (Exhibit A, lines 684 - 686) by calling KnowledgeAgents.Repository.transfuse() (Exhibit C, lines 197-297).

Claim 43	
The method according to claim 35, wherein adding the at least one of the found documents comprises removing one of the documents from the set responsive to adding the at least one of the found documents.	KnowledgeAgents.Repository.transfuse() "transfuses new good sites into the repository, replacing stale sites" (Exhibit C, line 198).
Claim 44	
The method according to claim 43, and comprising tracking a level of relevance of the reference documents to the queries, and wherein removing the one of the documents comprises removing one of the reference documents whose tracked level of relevance is low.	The class KnowledgeAgents.Repository tracks the relevance level for each of the sites in the repository in the SiteDB local object. (This object is declared in Exhibit C, line 17, and the scores are updated at lines 235-252.) These relevance levels are used to determine whether old sites will be kept or replaced during the update process performed by the KnowledgeAgents.Repository.transfuse() method, as described above.

Claim 45	
The method according to claim 35, wherein the corpus comprises at least a part of the World Wide Web, and the documents comprise Web pages, and wherein searching the corpus comprises conveying the query to one or more Web search engines.	KnowledgeAgents.Agent.textQuery() searches the World Wide Web (WWW, Exhibit A, lines 303-305). This method calls one or more search engines (line 315) to perform the search. In Exhibit E (page 16, last paragraph) we described the use of the AltaVista Web search engine in this manner.
Claim 46	
The method according to claim 45, wherein inputting the first query comprises receiving the query from a user of a pervasive device, and wherein searching the corpus comprises searching while the device is disconnected from the Web.	This feature may be implemented using the code in Exhibits A-C, but it is not explicitly shown in the exhibits. In regard to this claim, the Examiner indicated that Bowman et al. (U.S. Patent 6,006,225) would have led a person of ordinary skill in the art to carry out the step of searching while the device that received the search query is disconnected from the Web.

Claim 47	
The method according to claim 35, wherein identifying the set of reference documents comprises opening one or more files of a knowledge base on a computer in which data regarding the reference documents are saved.	This feature may be implemented using the code in Exhibits A-C, by applying the methods in the code to reference documents in a knowledge base on a computer rather than on the World Wide Web, but it is not explicitly shown in the exhibits. In regard to this claim, the Examiner indicated that Liddy et al. (U.S. Patent 6,304,864) would have led a person of ordinary skill in the art to open one or more files of a knowledge base on a computer in the context of identifying the set of reference documents as recited in this claim.

Claim 48	
The method according to claim 47, wherein identifying the set of reference documents comprises identifying the set of documents used by a first user in searching the corpus, and wherein opening the one or more files comprises copying the files for use by a second user in searching the corpus for information in the domain.	This feature may be implemented using the code in Exhibits A-C, by copying files opened by a first user for use by a second user, but it is not explicitly shown in the exhibits. In regard to this claim, the Examiner indicated that Liddy et al. (U.S. Patent 6,304,864) would have led a person of ordinary skill in the art to copy files opened by a first user for use by a second user in searching a corpus for information as recited in this claim.

5) Claims 51-57 and 59-61 recite apparatus and a computer software product, with limitations similar to those of certain of method claims 35-48. Based on the similarity of subject matter between the method, apparatus and software claims, it can similarly be demonstrated that we reduced to practice the entire invention recited in claims 51-57 and 59-61 prior to February 25, 2000.

6) We described the capabilities of our search software (as presented in Exhibits A-C) in the paper that is attached hereto as Exhibit E. As explained in section 4 of this paper, we defined knowledge agents in a number of different knowledge domains, including palm pilots, cryptography, artificial intelligence, geographic information systems, information

retrieval and Star Wars. The test results are described in detail on pages 14-18 of Exhibit E. The reported results demonstrate that our software successfully carried out the functions that are recited in the claims above.

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and conjecture are thought to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

Yariv Aridor
Citizen of Israel
31/b Yaalom Street, Zichron
Yaakov 30900
Israel

Date:

David Carmel
Citizen of Israel
12/5 Alexander Yanai Street,
Haifa 34816
Israel

Date:
January 23 2007

Michael Herscovici
Citizen of Israel
14 Got Levin Street, Haifa
32922
Israel

Date: 22 Jan 2007

Yoelle Maarek-Smadja
Citizen of Israel
[Address]
Israel

Date:

Aya Soffer
Citizen of Israel
33 Disraeli Street, Haifa
34333
Israel

Date:

Ronny Lempel
Citizen of Israel
1 Moshe Sneh Street, Haifa
34987
Israel

Date:

48760

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Application of : **ARIDOR et al.**

Serial No.: 10/634,319 : Group Art Unit: 2163

Filed : August 1, 2003 : Examiner: Hanh B. Thai

For : INFORMATION SEARCH USING KNOWLEDGE AGENTS

Honorable Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

DECLARATION UNDER 37 CFR 1.131

Sir:

We, the undersigned, Yariv Aridor, David Carmel, Michael Herscovici, Yoelle Maarek-Smadja, Aya Soffer and Ronny Lempel, hereby declare as follows:

1) We are the Applicants in the patent application identified above, and are the inventors of the subject matter described and claimed in claims 35-62 therein.

2) Prior to February 25, 2000, we reduced our invention to practice, as described and claimed in the subject application, in Israel, a WTO country. We implemented the invention in the form of software code in the Java programming language, and then tested the code successfully in a prototype system.

3) As evidence of the reduction to practice of the present invention, we attach hereto in Exhibits A-C parts of the Java source code that we used to implement the invention:

- Exhibit A: Class KnowledgeAgents.Agent
- Exhibit B: Class KnowledgeAgents.AgentGUI
- Exhibit C: Class KnowledgeAgents.Repository

A directory listing in Exhibit D (generated by the file archiving system used in the IBM Haifa Research Laboratory) shows the date on which the above source code files were stored on disk. The dates of the files, which are blacked out in Exhibit D, are prior to February 25, 2000. Results of testing this code are reported in a paper we prepared prior to February 25, 2000, entitled, "Knowledge Agents on the Web," which is attached hereto as Exhibit E.

4) Generally speaking, the software code in Exhibits A-C performs the functions of searching a corpus of documents, such as the World Wide Web, using knowledge agents that have developed specializations in certain knowledge domains. The following table shows the correspondence between the elements of method claims 35-48 in the present patent application and elements of the source code in Exhibits A-C:

Claim 35	**Source code**
A method for searching a corpus of documents, comprising:	KnowledgeAgents.AgentGUI and KnowledgeAgents.Agent provide a set of APIs to construct a domain specific agent and to use it for domain-specific search. See, for example, Exhibit A, lines 290-305.

defining a knowledge domain;	The KnowledgeAgents.Agent constructor (Exhibit A, lines 184-222) constructs a new agent for a given domain. The domain is defined by certain Web sites held in a repository (Exhibit C) that is specific to the agent/domain.
identifying a set of reference documents in the corpus pertinent to the domain;	The user initially specifies a set of sites (reference documents) using the "Add Sites" command in KnowledgeAgents.AgentGUI.actionPerformed() (Exhibit B, lines 445 - 450). This code calls the ActionAddSites sub-class defined in KnowledgeAgents.Agent (Exhibit A, lines 144 - 165), which adds the sites to the agent repository.
searching the corpus using the set of reference documents to find one or more of the documents in the corpus that contain information in the domain relevant to a first query; and	KnowledgeAgents.Agent.textQuery() (Exhibit A, lines 289-399) searches for a specific query over the agent's domain and finds sites that satisfy the query. The Boolean parameter *update* of this method (line 316) controls whether the search results will be added/updated into the agent's repository.

adding at least one of the found documents to the set of reference documents for use in searching the corpus for information in the domain relevant to a second, subsequent query, which is substantially different from the first query.	KnowledgeAgents.Agent.rankSites() (Exhibit A, lines 539-686) ranks the sites in which search results were found in the query stage. The repository is updated (lines 685-686) to include the best sites, i.e., sites with the highest weights. The updated repository is then available for use the next time textQuery() is invoked.
Claim 36	
The method according to claim 35, wherein inputting the first query comprises inputting one or more search terms.	The user inputs query terms as text strings using the "Refine Query" command in KnowledgeAgents.AgentGUI. actionPerformed() (Exhibit B, lines 436-441).

Claim 37	
The method according to claim 36, wherein searching the corpus comprises finding lexical characteristics of terms in the reference documents and refining the search terms using the lexical characteristics.	KnowledgeAgents.Agent.refineQuery() (Exhibit A, lines 1068-1087) receives as input a given query and number of terms to use in expanding the query. This method then expands the query (i.e., refines the search terms) using lexical characteristics (in the form of lexical affinities - "LAs") extracted from the agent's repository.
Claim 38	
The method according to claim 35, wherein inputting the first query comprises specifying one or more documents representative of the information to be found in the corpus.	KnowledgeAgents.Agent.linkQuery() (Exhibit A, lines 400-451) receives as input a set of documents (sites) that the user has specified as representing the information available on the Web. (The method returns an expanded set of sites that are optimally relevant to the given sites.)

Claim 39	
The method according to claim 35, wherein searching the corpus comprises searching the corpus to find the documents that contain the information relevant to the query and ranking the found documents by comparing them to the set of reference documents.	KnowledgeAgents.Agent.textQuery() calls KnowledgeAgents.Agent.rankSites() (Exhibit A, lines 539-687) which ranks the sites returned by the search according to the sites in the agent's repository. The comparison may be based on textual resemblance or on links, as noted with respect to claims 40 and 41 below.
Claim 40	
The method according to claim 39, wherein ranking the found documents comprises evaluating a textual resemblance between the found documents and the reference documents.	KnowledgeAgents.Agent.rankSites() ranks the search results by evaluating the textual resemblance between documents. It performs this function by calling (at line 595) KnowledgeAgents.Agent. normalizeTextWeights() (Exhibit A, lines 1213-1235).

Claim 41	
The method according to claim 39, wherein ranking the found documents comprises assessing links between the found documents and the reference documents.	KnowledgeAgents.Agent.getForwSet() finds all pages that the pages in the result set link to (Exhibit A, lines 729-868), while KnowledgeAgents.Agent.getBackSet() finds all pages that link to the result set (lines 689-728). Both methods are called by KnowledgeAgents.Agent.textQuery() in order to find all pages linking to and linked by the result page.
Claim 42	
The method according to claim 39, wherein adding the at least one of the found documents comprises adding at least the document having the highest ranking.	In KnowledgeAgents.Agent.rankSites(), after collecting all pages and scoring them, the top scored documents are used to update the agent's repository (Exhibit A, lines 684 - 686) by calling KnowledgeAgents.Repository.transfuse() (Exhibit C, lines 197-297).

Claim 43	
The method according to claim 35, wherein adding the at least one of the found documents comprises removing one of the documents from the set responsive to adding the at least one of the found documents.	KnowledgeAgents.Repository.transfuse() "transfuses new good sites into the repository, replacing stale sites" (Exhibit C, line 198).
Claim 44	
The method according to claim 43, and comprising tracking a level of relevance of the reference documents to the queries, and wherein removing the one of the documents comprises removing one of the reference documents whose tracked level of relevance is low.	The class KnowledgeAgents.Repository tracks the relevance level for each of the sites in the repository in the SiteDB local object. (This object is declared in Exhibit C, line 17, and the scores are updated at lines 235-252.) These relevance levels are used to determine whether old sites will be kept or replaced during the update process performed by the KnowledgeAgents.Repository.transfuse() method, as described above.

Claim 45	
The method according to claim 35, wherein the corpus comprises at least a part of the World Wide Web, and the documents comprise Web pages, and wherein searching the corpus comprises conveying the query to one or more Web search engines.	KnowledgeAgents.Agent.textQuery() searches the World Wide Web (WWW, Exhibit A, lines 303-305). This method calls one or more search engines (line 315) to perform the search. In Exhibit E (page 16, last paragraph) we described the use of the AltaVista Web search engine in this manner.
Claim 46	
The method according to claim 45, wherein inputting the first query comprises receiving the query from a user of a pervasive device, and wherein searching the corpus comprises searching while the device is disconnected from the Web.	This feature may be implemented using the code in Exhibits A-C, but it is not explicitly shown in the exhibits. In regard to this claim, the Examiner indicated that Bowman et al. (U.S. Patent 6,006,225) would have led a person of ordinary skill in the art to carry out the step of searching while the device that received the search query is disconnected from the Web.

Claim 47	
The method according to claim 35, wherein identifying the set of reference documents comprises opening one or more files of a knowledge base on a computer in which data regarding the reference documents are saved.	This feature may be implemented using the code in Exhibits A-C, by applying the methods in the code to reference documents in a knowledge base on a computer rather than on the World Wide Web, but it is not explicitly shown in the exhibits. In regard to this claim, the Examiner indicated that Liddy et al. (U.S. Patent 6,304,864) would have led a person of ordinary skill in the art to open one or more files of a knowledge base on a computer in the context of identifying the set of reference documents as recited in this claim.

Claim 48	
The method according to claim 47, wherein identifying the set of reference documents comprises identifying the set of documents used by a first user in searching the corpus, and wherein opening the one or more files comprises copying the files for use by a second user in searching the corpus for information in the domain.	This feature may be implemented using the code in Exhibits A-C, by copying files opened by a first user for use by a second user, but it is not explicitly shown in the exhibits. In regard to this claim, the Examiner indicated that Liddy et al. (U.S. Patent 6,304,864) would have led a person of ordinary skill in the art to copy files opened by a first user for use by a second user in searching a corpus for information as recited in this claim.

5) Claims 51-57 and 59-61 recite apparatus and a computer software product, with limitations similar to those of certain of method claims 35-48. Based on the similarity of subject matter between the method, apparatus and software claims, it can similarly be demonstrated that we reduced to practice the entire invention recited in claims 51-57 and 59-61 prior to February 25, 2000.

6) We described the capabilities of our search software (as presented in Exhibits A-C) in the paper that is attached hereto as Exhibit E. As explained in section 4 of this paper, we defined knowledge agents in a number of different knowledge domains, including palm pilots, cryptography, artificial intelligence, geographic information systems, information

retrieval and Star Wars. The test results are described in detail on pages 14-18 of Exhibit E. The reported results demonstrate that our software successfully carried out the functions that are recited in the claims above.

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and conjecture are thought to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

Yariv Aridor
Citizen of Israel
31/b Yaalom Street, Zichron
Yaakov 30900
Israel

Date:
Jan 23, 2007

David Carmel
Citizen of Israel
12/5 Alexander Yanai Street,
Haifa 34816
Israel

Date:

US 10/634,319

Declaration under 37 C.F.R 1.131 by Aridor et al.

Michael Herscovici
Citizen of Israel
14 Got Levin Street, Haifa
32922
Israel

Date:

Yoelle Maarek-Smadja
Citizen of Israel
[Address]
Israel

Date:

Aya Soffer
Citizen of Israel
33 Disraeli Street, Haifa
34333
Israel

Date:
Feb. 13 2007

Ronny Lempel

Ronny Lempel
Citizen of Israel
1 Moshe Sneh Street, Haifa
34987
Israel

Date:
Jan 23, 2007